

Golfcar and ROS

Golfcar: Yamaha G22E GMAX 48V

Robot **O**perating **S**ystem

Please check out the codes and one recorded data from
<https://svn.csail.mit.edu/karaman/code/fm>

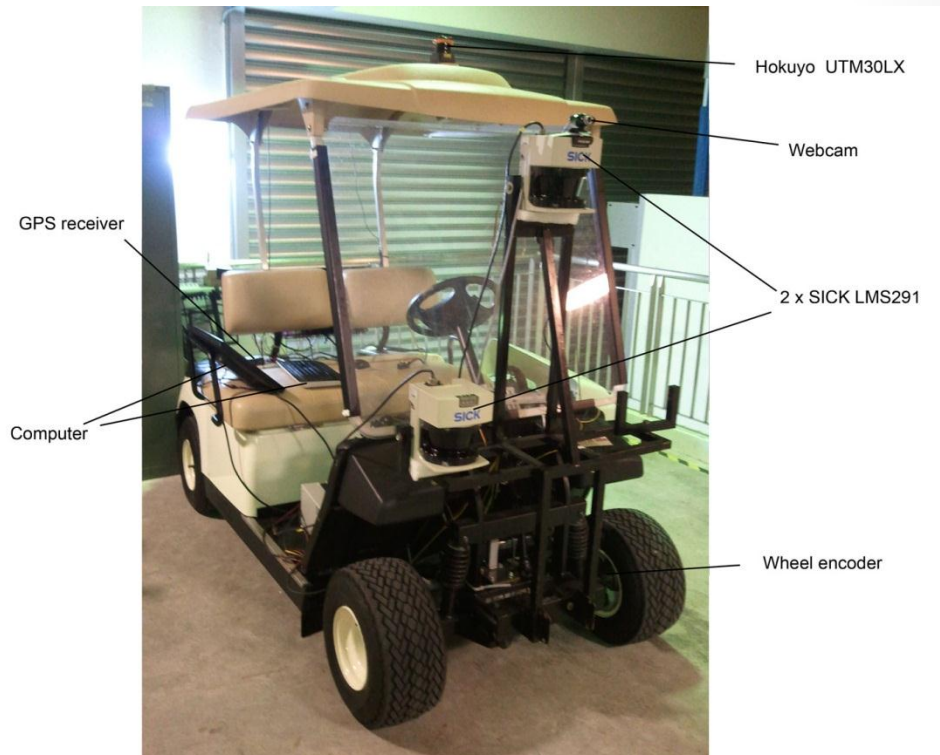
Current hardware

Platform

- Yamaha Gmax Golf cart with drive by wire capability
 - Running Ubuntu with ROS

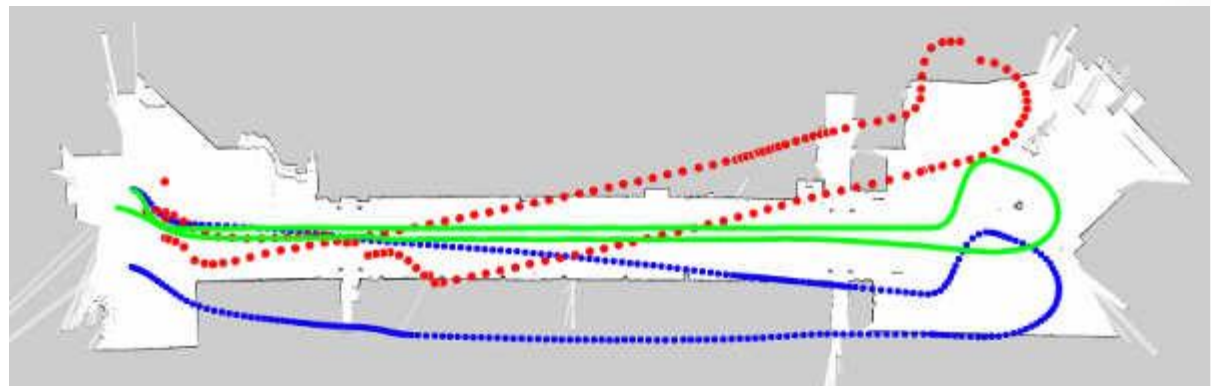
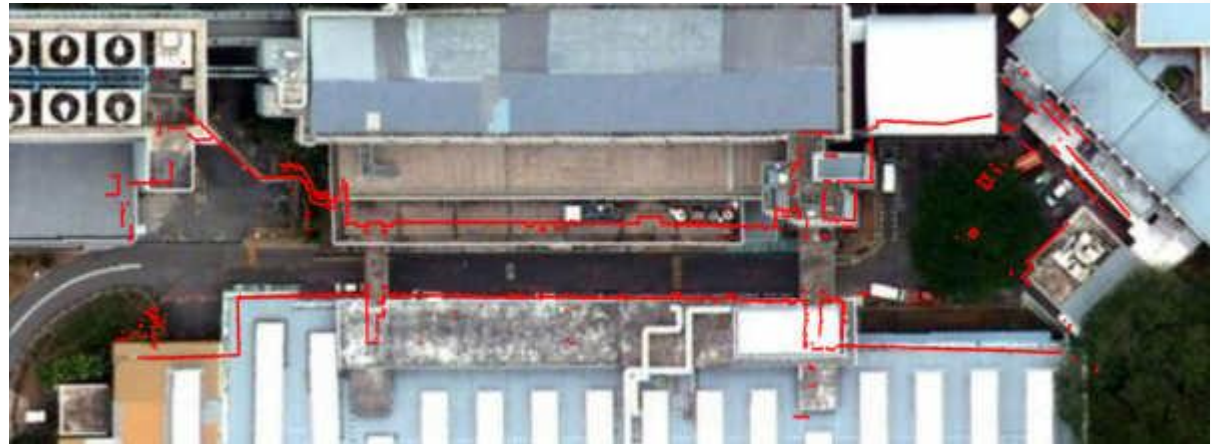
Sensors

- Wheel and steering encoders
- GPS + Gyro
 - Ublox AEK-6R evaluation kit with dead reckoning
 - Integrated EKF
- Laser range finders
 - SICK-LMS 291
 - HOKUYO UTM 30LX
- Vision system



Localization

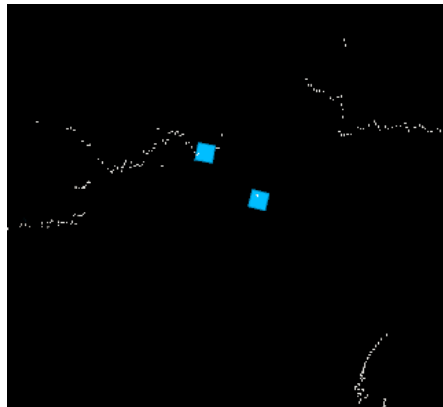
- Laser based mapping, Monte Carlo Localization



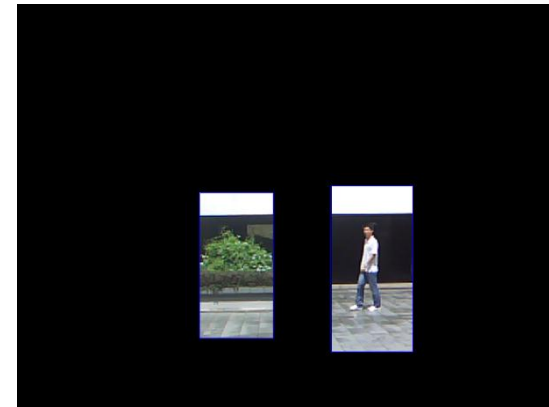
Pedestrian Detection



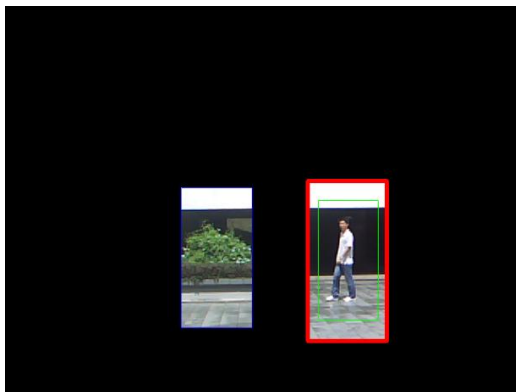
(a) Camera Input



(b) Laser Input



(c) laser based image segmentation



(d) HOG classifier output



(e) Robust pedestrian detection



(f) Track of single pedestrian.



Exploiting Infrastructure Sensors



(a) Onboard camera view



(b) Infrastructure camera view



(c) Vehicle in operation



pedestrianVisual2.mp4

ROS

- We are using
 - ROS latest distribution release: Diamondback
 - Ubuntu 10.04 LTS

 ROS.org



Overview

- ROS Introduction
- ROS tutorial
 - Topics
 - Services
 - Actions
 - Messages
 - Parameters
 - Packages and Stacks
- Examples

Materials obtained from:

<http://www.ros.org/wiki/ROS/Tutorials>

<http://www.ros.org/wiki/Events/CoTeSys-ROS-School>

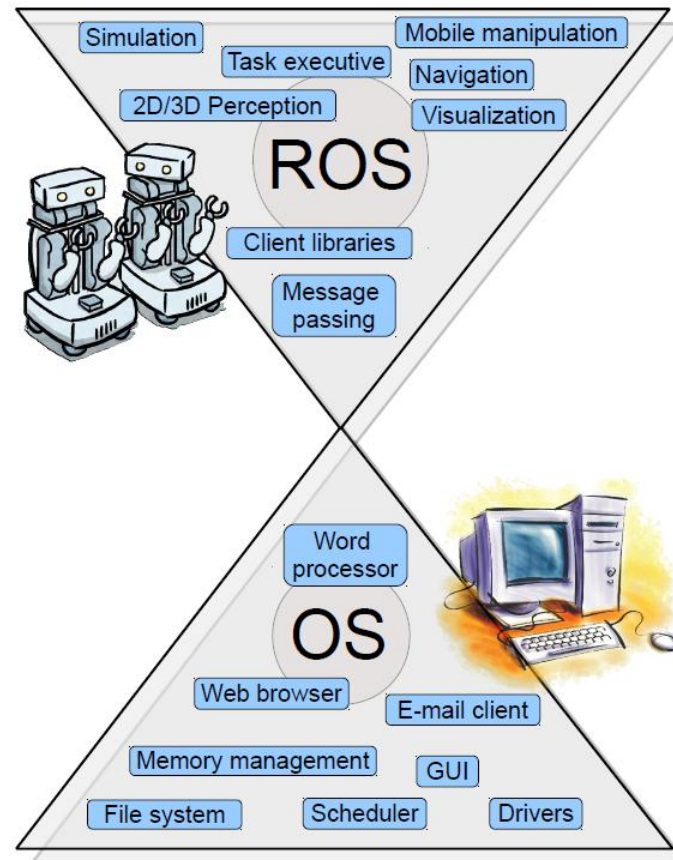
ROS Introduction

- What is ROS?
 - Meta operating system for robotics
 - Obtain, build, write, and run code across multiple computers



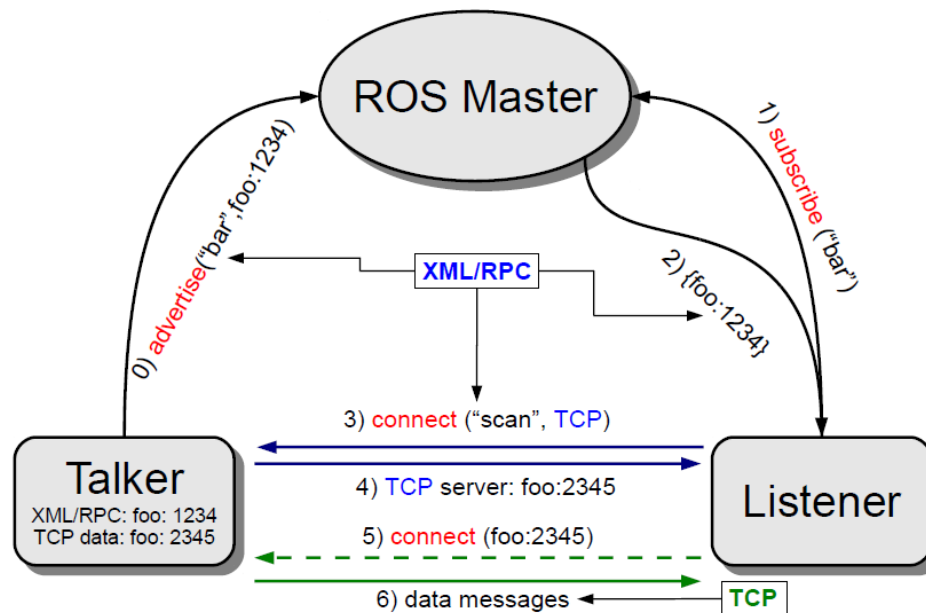
ROS Introduction

- High-level view



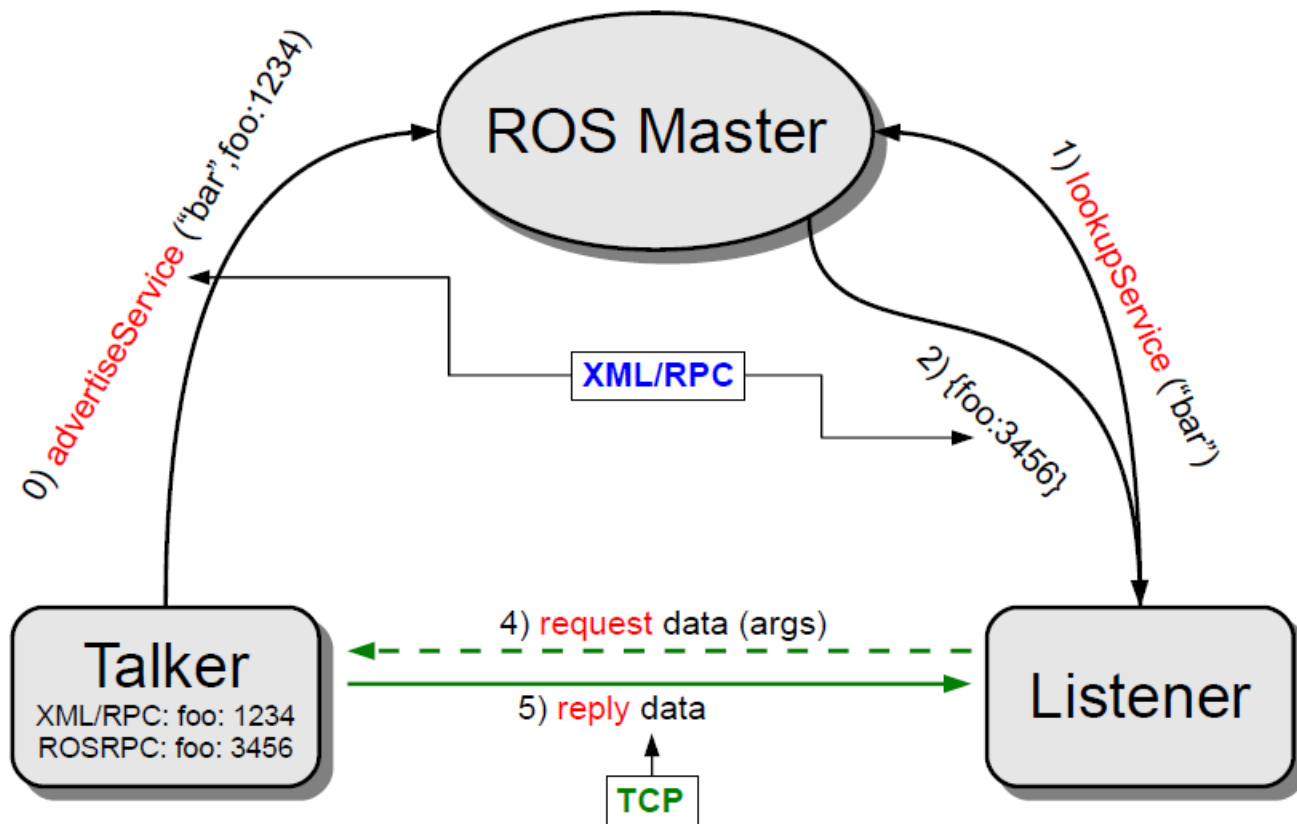
ROS Tutorial

- Topics
 - Nodes connect via topics
 - The discovery of who publishes on what topic is done via a ROS master



ROS Tutorial

- Services

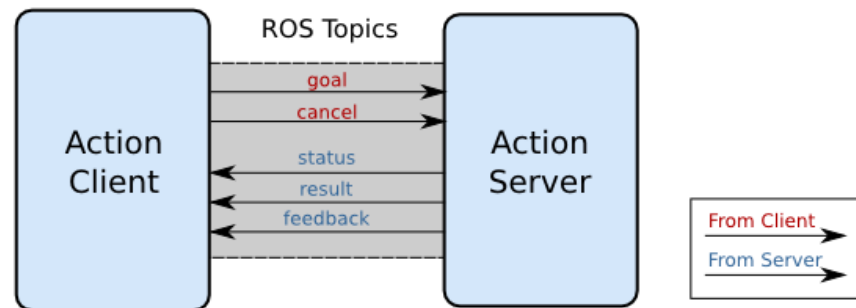


ROS Tutorial

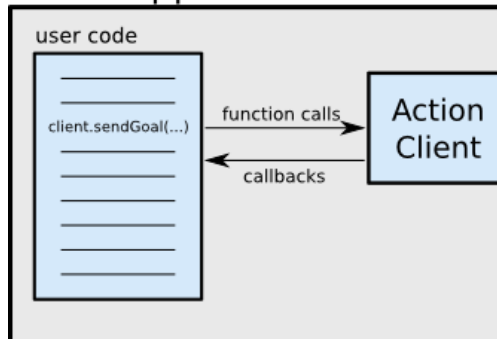
- Actions

- Using function calls and callbacks
 - Request goals
 - Execute goals
- Action protocol relies on ROS topics to transport messages

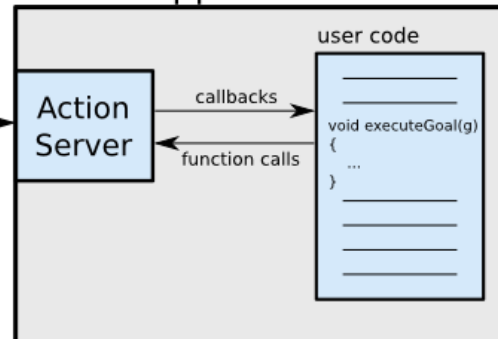
Action Interface



Client Application



Server Application



ROS

ROS Tutorial

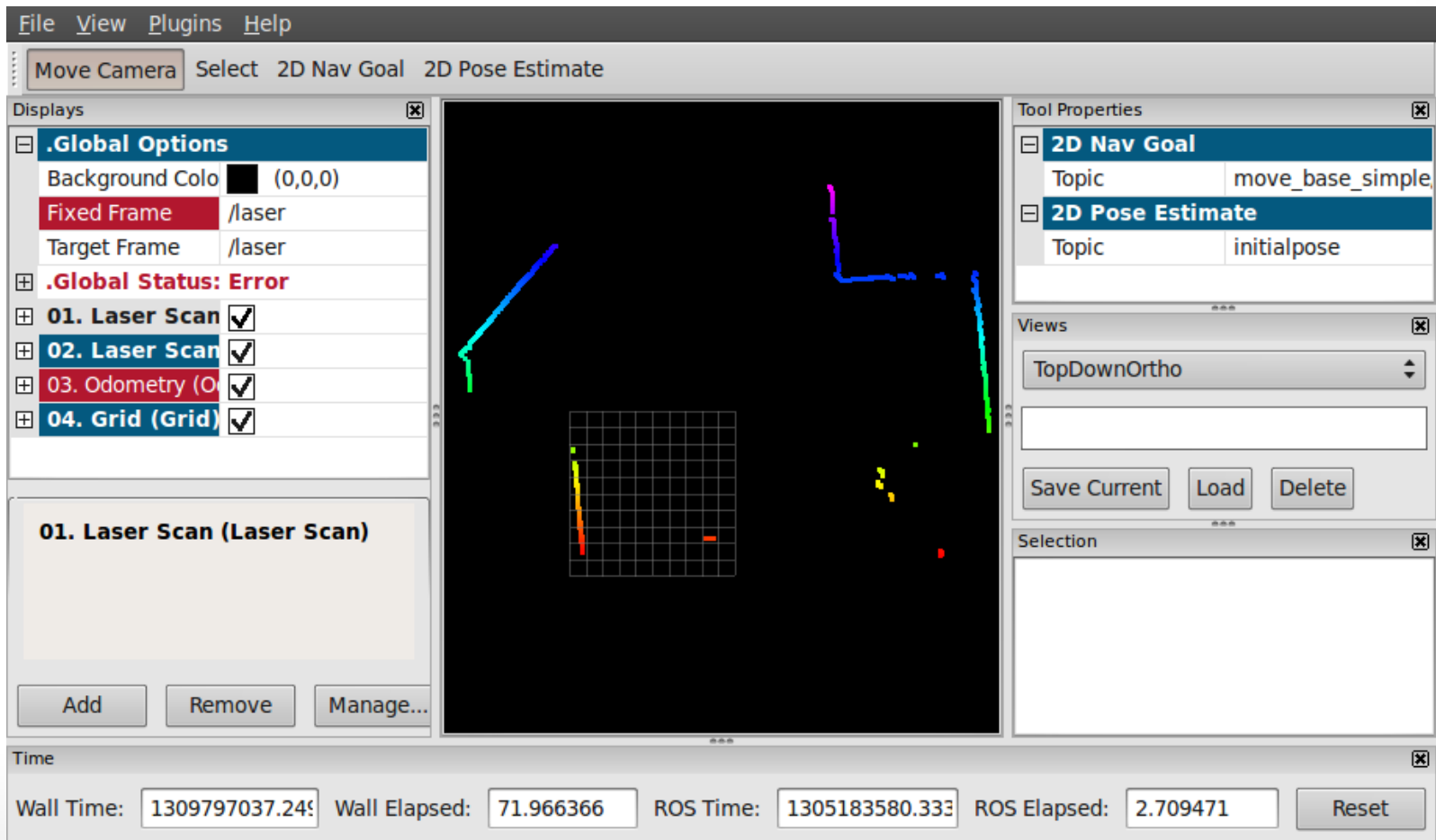
- Parameters
 - Nodes can set parameters on the server
 - Any other nodes can read them
 - Have unique names
 - Can represent primitive data types
 - Integers
 - Floats
 - Boolean
 - ...etc
 - Can be set and remapped at runtime
 - Stored on the parameter server

ROS Tutorial

- Packages and Stacks
 - Packages: directories with a certain structure, can contain anything: nodes, messages, tools
 - In their most basic form:
 - Package_name
 - Package_name/Makefile
 - Package_name/CMakeLists.txt
 - Package_name/manifest.xml
 - Stacks: collection of packages
 - Stack_name
 - Stack_name/package_name_1
 - Stack_name/package_name_N
 - Stack_name/stack.xml

Examples 1

- Playback ROS bag and get familiarize with rviz



Example 1

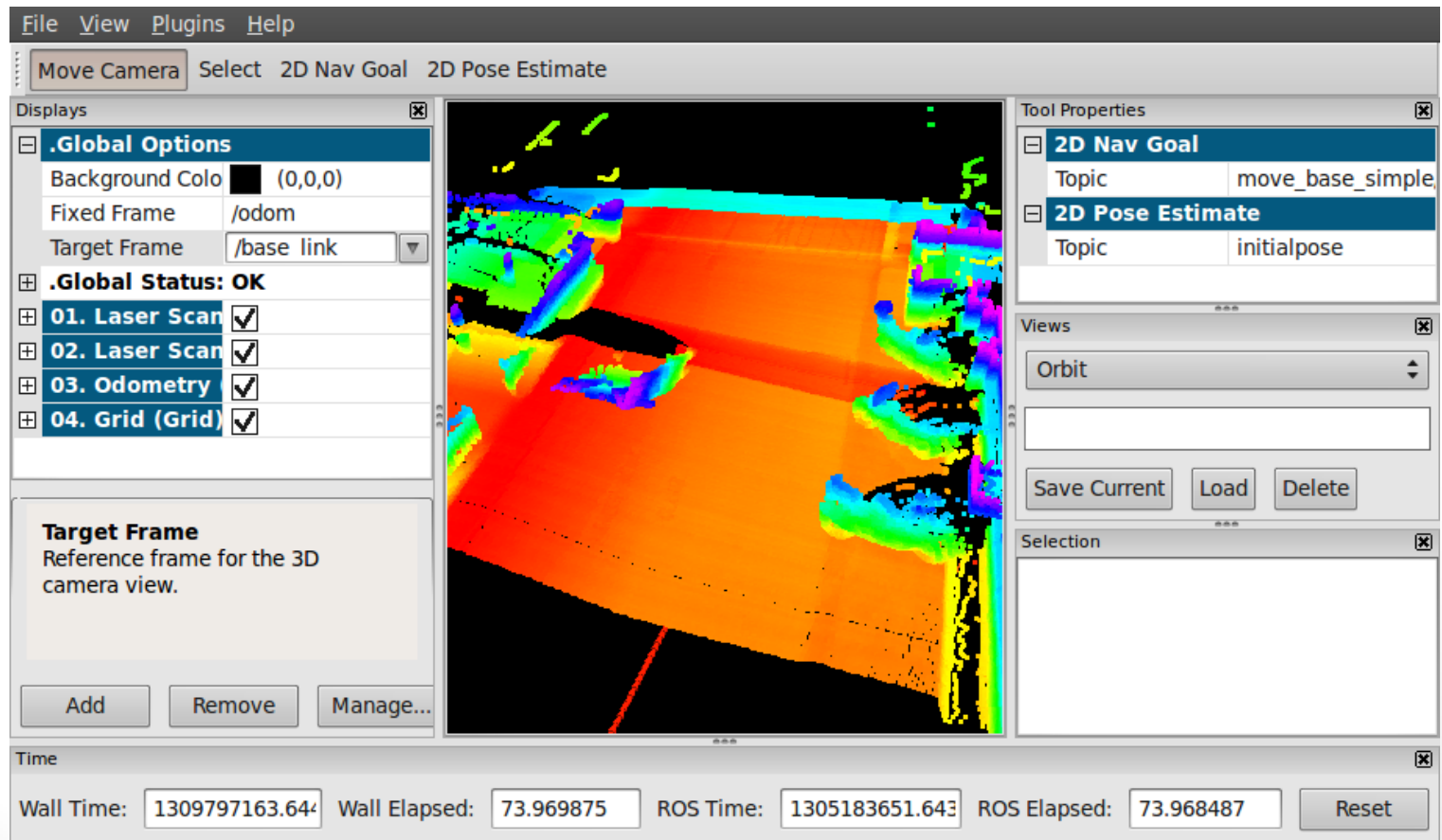
- Some pre-requisite
 - Check out the codes from svn
 - svn co <https://svn.csail.mit.edu/karaman/code/fm>
 - Edit ROS_PACKAGE_PATH in /opt/ros/diamondback/setup.sh to the following (Change "~.../smart-ros-pkg" to the correct directory):
 - export ROS_PACKAGE_PATH=/opt/ros/diamondback/stacks:~.../smart-ros-pkg
 - Update the environment of the current terminal
 - source /opt/ros/diamondback/setup.bash

Examples 1

- Playback ROS bag and get familiarize with rviz
 - To start up ROS master, parameter server and roscore logging node
 - `roscore`
 - To run at bag file's time stamp
 - `rosparam set_sim_time true`
 - To run ROS visualization
 - `roslaunch rviz rviz`
 - To play the recorded data
 - `roslaunch play tutorial.bag --clock`

Examples 2

- Running a package together with ROS bag, and the use of tf with roslaunch

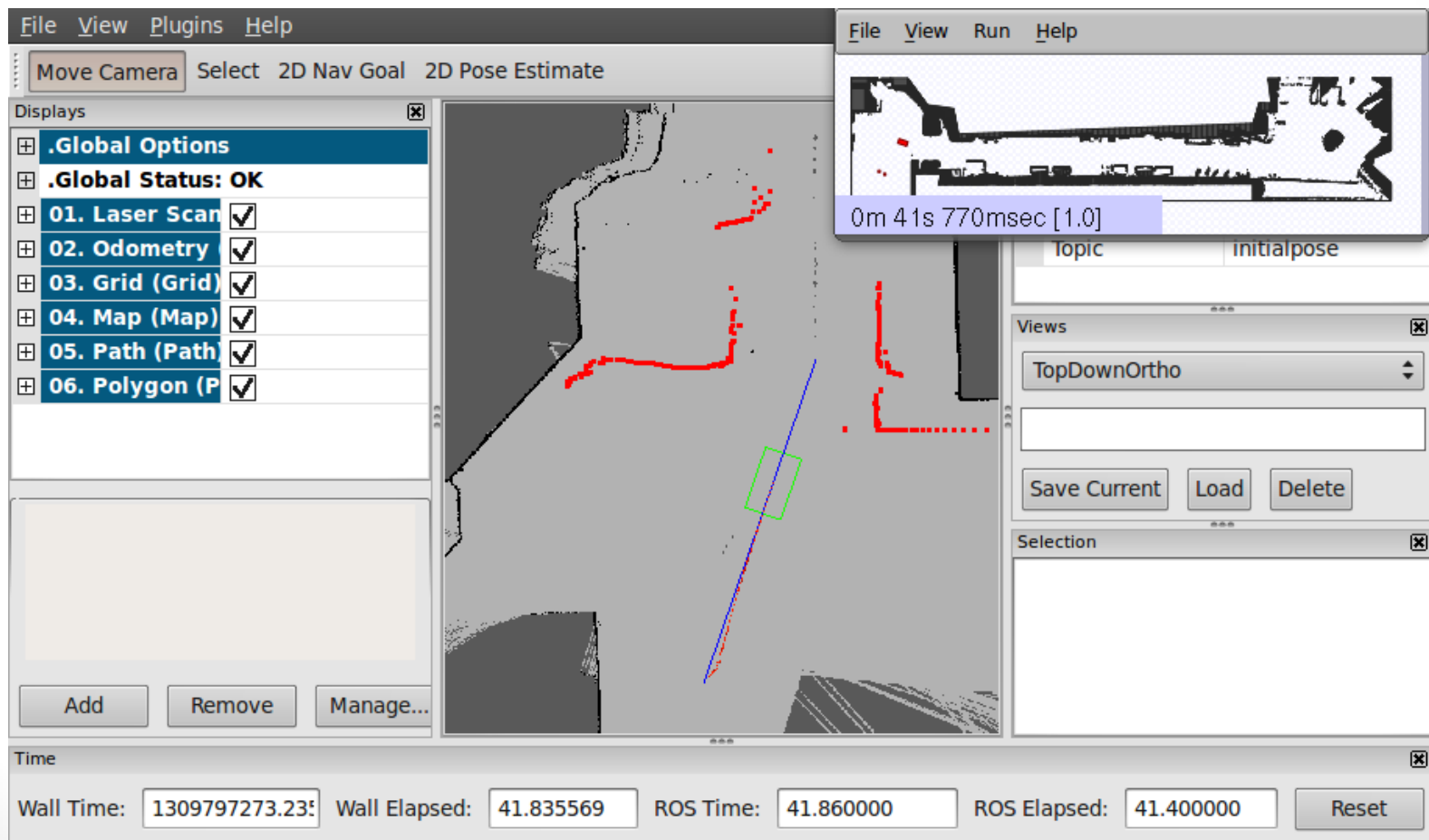


Examples 2

- Running a package together with ROS bag, and the use of tf with roslaunch
 - rosmake under the smart-ros-pkg stack folder
 - All packages is now compiled, to run odometry package
 - rosrn golfcar_odom golfcar_odom_imu
 - Transform is required to tell ROS the placement of different sensors
 - roslaunch Launch tf.launch
 - To play the recorded data
 - rosbag play tutorial.bag -clock
 - Change the Fixed Frame and Target Frame under rviz's .Global Options
 - Select odom as fixed frame and base_link as target frame
 - Feel free to change the Views from Top Down to Orbit and try to navigate around the visualizer with mouse

Examples 3

- Golfcar Navigation Simulation



Examples 3

- Golfcar Navigation Simulation
 - Close all the previous running processes
 - To run the complete simulation, simply
 - `roslaunch Launch simulation.launch`
 - To move the simulated vehicle
 - Under rviz,
 - Click on 2D Nav Goal button
 - Click on any place on the map to tell the `move_base` package where to go